תרגיל הרצה 1 –

יהב יהושוע בריח (326295417) ואיתי יהודה בנדרסקי (326621927)

מבנה הרשת:

1. שכבת BTU - שכבת הפעלת סיגמויד מותאם אישית – במקרה שלנו temp = 0.001.
2. שכבת רשת - (Network) - רשת הכוללת:
   - שכבה נסתרת (hidden layer) - שכבה הקיימת בין הקלטים לבין השכבה החיצונית. מכילה מספר נוירונים שנקבע על ידי המשתמש (K).
   - שכבה חיצונית (output layer) - שכבה הכוללת נוירון אחד אשר פולט את התוצאה הסופית. במידה והמשתמש בוחר להפעיל את bypass - חישוב הפלט יכלול גם את הקלט המקורי בנוסף לפלט של השכבה הנסתרת.

K = 1

בחרנו את המשקולות והביאסים כך שבשכבה הנסתרת – הנוירון מבצע AND ובכך בודק האם המצב בקלט הוא [1,1] – אם כן אז מוכפל במשקל 2- ולכן הנוירון החיצוני יפלוט 0. כדי לטפל בשאר מצבי הקלט בצעתי Bypass שבאמצעותו ובחיבור עם הביאס של הנוירון החיצוני – הנוירון החיצוני יפלוט 1 אם לפחות אחד מהקלטים הוא 1 (זה אומר שהקלטים הם [0,1] או [1,0]), אחרת יפלוט גם 0. לכן יפלוט 1 כאשר הקלטים הם [0,1] או [1,0] ו0 כאשר הקלטים הם [1,1] או [0,0].

פלט –

```
Please choose the number of hidden neuron (1/2/4), 0 for end: 1
Layer: hidden.weight -> Values: [[1. 1.]]
Layer: hidden.bias -> Values: [-1.5]
Layer: output.weight -> Values: [[ 1.  1. -2.]]
Layer: output.bias -> Values: [-0.5]

The loss is: 0.0

Truth Table:
Input (x): [0. 0.] -> Output (y): 0.0000
Input (x): [0. 1.] -> Output (y): 1.0000
Input (x): [1. 0.] -> Output (y): 1.0000
Input (x): [1. 1.] -> Output (y): 0.0000
```

K = 2

בחרנו את המשקולות והביאסים כך שבשכבה הנסתרת - הנוירון השמאלי יהיה אחראי לבדוק האם בקלטים יש לפחות 0 אחד, והנוירון הימני אחראי לבדוק האם בקלטים יש לפחות 1 אחד. בכך כאשר בקלטים יהיה לפחות 0 אחד ו1 אחד – זה אומר שהקלטים הם [0,1] או [1,0] – הנוירונים בשכבה הנסתרת יפלטו את הערך 1 (כל אחד מהם) והניורון של השכבה החיצונית מבצע את פעולת AND, ולכן יפלוט 1 כאשר הקלטים הם [0,1] או [1,0] ו0 כאשר הקלטים הם [1,1] או [0,0].

פלט –

```
Please choose the number of hidden neuron (1/2/4), 0 for end: 2
Layer: hidden.weight -> Values: [[-1. -1.]
 [ 1.  1.]]
Layer: hidden.bias -> Values: [ 1.5 -0.5]
Layer: output.weight -> Values: [[1. 1.]]
Layer: output.bias -> Values: [-1.5]

The loss is: 0.0

Truth Table:
Input (x): [0. 0.] -> Output (y): 0.0000
Input (x): [0. 1.] -> Output (y): 1.0000
Input (x): [1. 0.] -> Output (y): 1.0000
Input (x): [1. 1.] -> Output (y): 0.0000
```

K = 4

בחרנו את המשקולות והביאסים כך שבשכבה הנסתרת כל נוירון אחראי על מצב אחד אפשרי של הקלטים ([0,0], [0,1], [1,0], [1,1]), ולכן בכל מצב רק נוירון אחד בשכבה הנסתרת יכול לפלוט 1, והנוירון בשכבה החיצונית יפלוט 1 כאשר וקטור הפלט מהשכבה הנסתרת הוא [0,1,0,0] או [0,0,1,0], ולכן יפלוט 1 כאשר הקלטים הם [0,1] או [1,0] ו0 כאשר הקלטים הם [1,1] או [0,0].

פלט –

```
Please choose the number of hidden neuron (1/2/4), 0 for end: 4
Layer: hidden.weight -> Values: [[-1. -1.]
 [-1.  1.]
 [ 1. -1.]
 [ 1.  1.]]
Layer: hidden.bias -> Values: [ 0.5 -0.5 -0.5 -1.5]
Layer: output.weight -> Values: [[0. 1. 1. 0.]]
Layer: output.bias -> Values: [-0.5]

The loss is: 0.0

Truth Table:
Input (x): [0. 0.] -> Output (y): 0.0000
Input (x): [0. 1.] -> Output (y): 1.0000
Input (x): [1. 0.] -> Output (y): 1.0000
Input (x): [1. 1.] -> Output (y): 0.0000
```

מסקנות:

- היכולת להגדיר את המשקולות והביאסים באופן ידני מאפשרת למשתמש להבין איך כל שינוי במבנה של הרשת משפיע על הפלט שלה.
- למרות שהקוד לא כולל אלגוריתם למידה כמו gradient descent (שמתעדכן באופן אוטומטי), המשתמש יכול לעדכן את המשקולות והביאסים כדי לראות איך זה משפיע על תוצאות הרשת.
- כל בחירה במשקולות ובביאסים משפיעה על איך הרשת תבצע את החישוב, ובמיוחד כאשר מדובר בבעיית XOR, המשקולות והביאסים חייבים להיות מותאמים כדי שהרשת תוכל ללמוד את התבנית הנדרשת.
- המודל יכול להיות משולב עם אלגוריתמים אופטימיזציה בעתיד, כך שהוא ילמד אוטומטית את המשקולות והביאס מתוך הנתונים.